TWO DIMENSIONAL ANALYSIS OF FLUID-STRUCTURE INTERACTION BY METHOD OF FINITE DIFFERENCES - HYDRAULIC RAM, THE FUEL TANK PROBLEM

John Charles Bitzberger

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

TWO DIMENSIONAL ANALYSIS
OF FLUID-STRUCTURE INTERACTION
BY METHOD OF FINITE DIFFERENCES –
HYDRAULIC RAM, THE FUEL TANK PROBLEM

by

John Charles Bitzberger

June 1974

Thesis Advisor: R.E. Ball

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Two Dimensional Analysis of Fluid-Structure Interaction by Method of Finite Differences — Hydraulic Ram, The Fuel Tank Problem		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1974
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John Charles Bitzberger		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1974
		13. NUMBER OF PAGES 72
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fluid-Structure Interaction
Hydraulic Ram

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

As part of an ongoing research program at the Naval Postgraduate Schsol concerned with aircraft vulnerability and, in particular, the "Hydraulic Ram" phenomenon, two Fortran IV computer codes were written to analyze in two dimensions problems concerning structure-fluid interaction. This thesis

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. ABSTRACT continued)

presents the method selected for solution, details the codes written and presents test cases and example problems. User's instructions and program listings for both codes are also included.

Two Dimensional Analysis
of Fluid-Structure Interaction
by Method of Finite Differences –
Hydraulic Ram, The Fuel Tank Problem

by

John Charles Bitzberger
Lieutenant, United States Navy
B.S., Naval Postgraduate School, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1974

ABSTRACT

As part of an ongoing research program at the Naval Postgraduate School concerned with aircraft vulnerability and, in particular, the "Hydraulic Ram" phenomenon, two Fortran IV computer codes were written to analyze in two dimensions problems concerning structure-fluid interaction. This thesis presents the method selected for solution, details the codes written and presents test cases and example problems. User's instructions and program listings for both codes are also included.

TABLE OF CONTENTS

I. INTRODUCTION

Many important structural problems in aerospace engineering involve components that are in contact with a fluid. One example is the fuel tank located in a wing or fuselage of an aircraft. Another example is the fuel tank in a liquid fueled rocket. An example in the field of civil engineering is the dam and reservoir system.

An ongoing research program at the Naval Postgraduate School is concerned with aircraft vulnerability in the fuel tank area. This program is currently devoted to a study of the Hydraulic Ram Problem. "Hydraulic Ram" is the dynamic loading of fuel tanks when impacted by bullets or warhead fragments. This loading gives the appearance of a simple pressure pulse with a resulting deformation or rupture of the tank. It has been found, however, that the loading and deformation are a combination of a number of events called "hydraulic ram components." [Ref. 1]

As part of the theoretical analysis of the fuel tank response to a penetrating projectile two computer codes were written to solve for the response of the fluid and the walls of a idealized two-dimensional fuel tank. The codes will compute the pressure throughout the fluid and will calculate the wall response to this pressure. The user can prescribe the problem to be analyzed, including, but not limited to:

Fluid surface pressure disturbances, initial wall displacements, initial wall velocities, pressure distributions through the fluid, etc.

This thesis presents the method selected for the solution of problems involving two-dimensional fluid-structure interaction. The types of problems which can be solved using the codes are described. Section II contains the theoretical description of the problem, and the following two sections outline two finite difference formulations that were used to obtain solutions. Section V contains a description of each code and describes the modifications necessary to change the boundary and initial conditions. Section VI presents the solutions obtained with the codes, including the check cases and several example programs, and discusses the comparison of results obtained from the two code formulations. Users' instructions and program listings for both formulations are included in Appendices A and B.

II. DERIVATION OF GOVERNING EQUATIONS

A. THE WAVE EQUATION FOR AN INVISCID FLUID

1. Equation of Motion

Euler's equation of motion can be given in the form

$$\rho \frac{D\bar{q}}{dt} + \text{grad } p = 0 \qquad (1)$$

where ρ is the fluid density, p is the fluid hydrostatic pressure (positive in compression), $\bar{q}$ is the velocity vector and t is time. The total derivative $\frac{D}{Dt}$ is given by

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial t} \qquad (2)$$

where u, v, w are the components of the velocity in the x, y, z directions respectively. When the fluid velocity is sufficiently small,

$$\frac{D}{Dt} \approx \frac{\partial}{\partial t} \qquad (3)$$

Superscript dots will be used to denote partial derivatives with respect to time.

2. Constitutive Equation

The constitutive equation of the inviscid fluid is given by

$$\dot{p} = - K \text{ div } \bar{q} \qquad (4)$$

where K is the bulk modulus.

3. <u>Equation of Continuity</u>

$$\frac{1}{\rho}\frac{D\rho}{Dt} + \text{div}\ \bar{q} = 0 \qquad (5)$$

Applying Eq. (3) we obtain

$$\frac{1}{\rho}\dot{\rho} + \text{div}\ \bar{q} = 0 \qquad (6)$$

Eliminating div $\bar{q}$ between Eqs. (4) and (6) gives

$$\frac{\dot{p}}{K} = \frac{1}{\rho}\dot{\rho} \qquad (7)$$

4. <u>Potential Equation</u>

Define a potential

$$\bar{q} \triangleq \text{grad}\ \phi \qquad (8)$$

Thus Eq. (1) becomes

$$\rho\ \text{grad}\ \dot{\phi} + \text{grad}\ p = 0 \qquad (9)$$

when Eq. (3) is applied. Assuming ρ to be essentially constant throughout the fluid, Eq. (9) becomes

$$\text{grad}\ (\rho\dot{\phi} + p) = 0 \qquad (10)$$

or

$$\rho\dot{\phi} + p = \text{constant} \qquad (11)$$

The constant term is set to zero since it implies a base pressure which is constant throughout the domain.

Differentiating Eq. (1) with respect to time gives

$$\rho\ddot{\bar{q}} + \text{grad}\ \dot{p} = 0 \tag{12}$$

Applying Eqs. (4) and (8)

$$\rho\ \text{grad}\ \ddot{\phi} + \text{grad}\ (-K\ \text{div grad}\ \phi) = 0 \tag{13}$$

Since ρ is assumed to be essentially constant, Eq. (13) becomes

$$\text{grad}(\rho\ddot{\phi} - K\ \text{div grad}\ \phi) = 0 \tag{14}$$

or

$$\text{grad}(\rho\ddot{\phi} - K\nabla^2\phi) = 0 \tag{15}$$

Then

$$\rho\ddot{\phi} - K\nabla^2\phi = \text{constant} \tag{16}$$

or

$$\rho\ddot{\phi} = K\nabla^2\phi \tag{17}$$

Recalling that the speed of sound in a fluid can be determined from the relationship

$$c^2 = \frac{K}{\rho} \tag{18}$$

we obtain the wave equation

$$c^2\nabla^2\phi = \ddot{\phi} \tag{19}$$

B. THE GOVERNING EQUATION FOR THE STRUCTURE

1. <u>Two-Dimensional Idealization</u>

Consider a long open box composed of five uniform plates. It can be idealized in two dimensions as a structure composed of three unit width beams attached at right angles as shown in Fig. 1. In the analysis each beam is considered separately, with its own set of independent boundary conditions. In all of the examples considered, each beam was assumed to be clamped at both ends.

The equation for the lateral displacement of a uniform beam under dynamic load is [Ref. 2]

$$EI \frac{\partial^4}{\partial x^4} w(x,t) + m\ddot{w}(x,t) = - p \tag{20}$$

where E is Young's modulus, I is the section moment of inertia, m is the mass per unit length, w(x,t) is the lateral displacement of the beam, and p is the pressure applied to the beam. A positive pressure is in the opposite direction as w(x,t) as shown in Fig. 1 for the tank.

C. INTERFACE CONDITIONS

From Eq. (11) the pressure on the beam is the fluid pressure given by

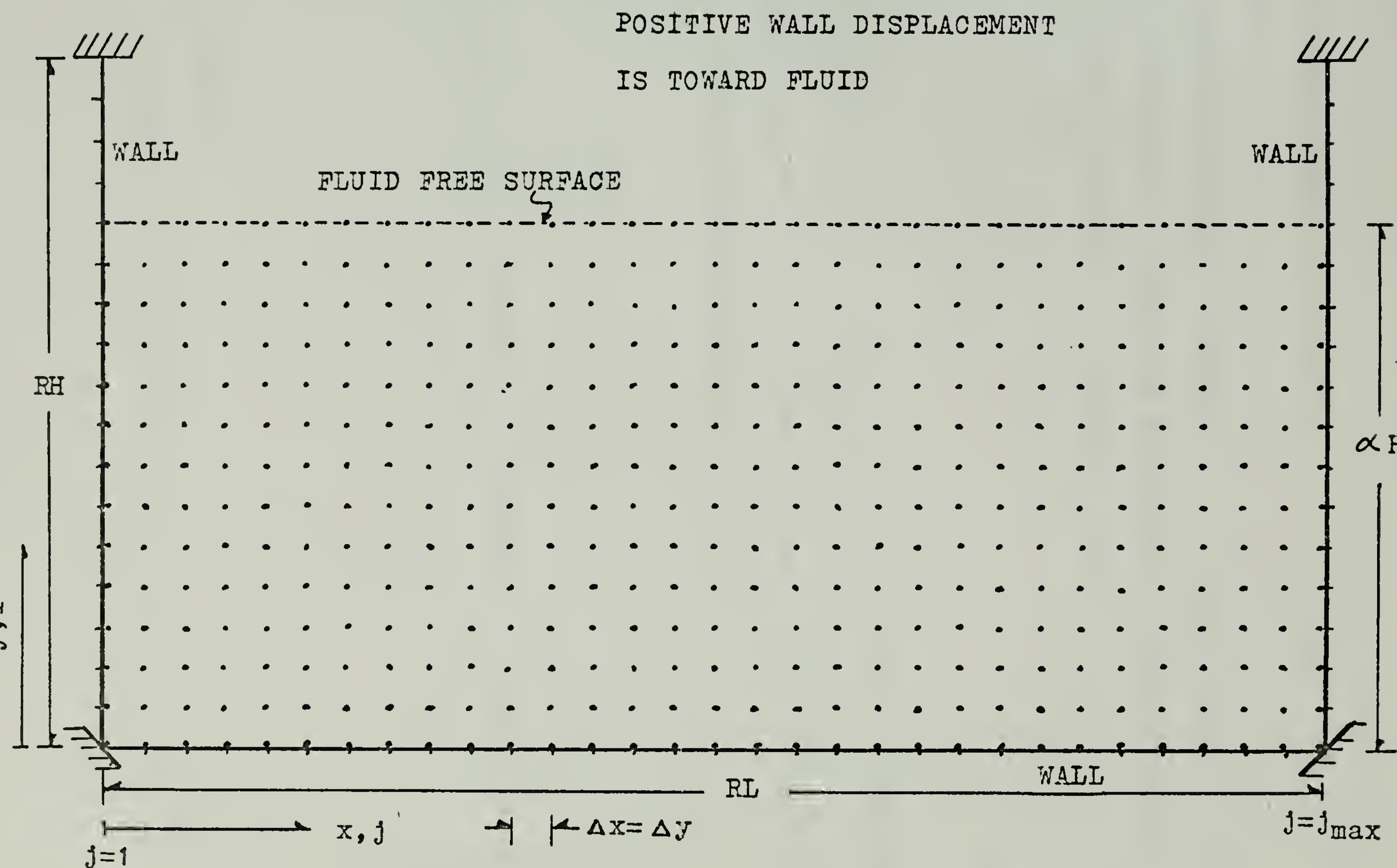

POSITIVE WALL DISPLACEMENT
IS TOWARD FLUID
WALL
WALL
FLUID FREE SURFACE
RH
αH
y, i
RL
WALL
x, j
$\Delta x = \Delta y$
$j=1$
$j=j_{max}$

FIG. 1

$$p = -\rho\dot{\phi} \tag{21}$$

where again a positive pressure is in the opposite direction as $w(x,t)$.

Another interface condition is the requirement that the normal velocity of the fluid potential at the interface is equal to the velocity of the wall, i.e.

$$\dot{w}(x,t) = \frac{\partial}{\partial \hat{n}} \phi \tag{22}$$

where $\frac{\partial}{\partial \hat{n}}$ indicates a partial derivative normal to the wall.

Where the fluid has a free surface and the pressure is zero, the boundary condition is given by

$$\phi = 0 \tag{23}$$

When an excitation pressure is applied to the free surface

$$\dot{\phi} = -\frac{p}{\rho} \tag{24}$$

D. SUMMARY

In summary, the equations used in the analysis are

$$c^2\nabla^2\phi = \ddot{\phi} \tag{19}$$

$$EI \frac{\partial}{\partial x^4} w(x,t) + m\ddot{w}(x,t) = -p \tag{20}$$

$$p = - \rho\dot{\phi} \tag{21}$$

$$\dot{w}(x,t) = \frac{\partial}{\partial \hat{n}} \phi \tag{22}$$

III. NUMERICAL METHOD OF ANALYSIS – ACCELERATION FORMULATION

A. FINITE DIFFERENCING SCHEME

1. The Fluid

The wave equation is hyperbolic in nature and for this reason an explicit formulation for the timewise response is chosen. The conventional central finite difference equations are used in both space and time, and when applied to Eq. (19) give [Ref. 3]

$$\frac{c^2}{(\Delta x)^2}\left\{\phi_{i,j-1} + \phi_{i,j+1} - 4\phi_{i,j} + \phi_{i+1,j} + \phi_{i-1,j}\right\}_k =$$

$$\left(\frac{1}{\Delta t}\right)^2 \left\{\phi_{k-1} - 2\phi_k + \phi_{k+1}\right\}_{i,j} \tag{25}$$

where $\Delta x = \Delta y$ in a square mesh, i and j are mesh point reference indices, and k is the time step. $\phi_{i,j,k+1}$ is the only unknown. All other terms are known or can be evaluated prior to the calculation of $\phi_{i,j,k+1}$.

2. The Beam Equation

Conventional central finite difference schemes are also used for the walls in both space and time. The resulting equation is given by [Ref. 3]

$$\frac{EI}{(\Delta x)^4}\left\{w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2}\right\}_k +$$

$$\frac{m}{(\Delta t)^2}\left\{w_{k-1} - 2w_k + w_{k+1}\right\}_i = -p_{i,k} \qquad (26)$$

where

$$p_{i,k} = -\frac{\rho}{2\Delta t}\left\{-\phi_{k-1} + \phi_{k+1}\right\}_{i,j} \qquad (27)$$

and $j = 1$ or $j = j_{max}$. See Fig. 1.

A similar pair of equations can be written for the wall along the boundary where $i = 1$. In Eq. (26) $w_{i,k+1}$ is the only unknown in the explicit scheme. Note that the pressure given by Eq. (27) is a function of the fluid potential at the wall.

3. The Calculation Scheme

When the two systems are combined, the other linking mechanism is the interface condition on the fluid and wall velocities. Using central finite differences, Eq. (22) is given by [Ref. 3]

$$\frac{1}{2\Delta t}\left\{-w_{i,k-1} + w_{i,k+1}\right\} = \frac{1}{2\Delta x}\left\{-\phi_{i,j-1,k} + \phi_{i,j+1,k}\right\} \qquad (28)$$

In the calculation scheme the wall deflection at each new step k+1 is computed first, and then the velocity potential is computed throughout the fluid. In order to compute the wall displacement at k+1 the pressure must be known at the preceding time step k, which requires knowledge of the velocity potential at the wall at the new time step, k+1, according to Eq. (27). The solution to this problem is to solve for the wall displacement by implicitly solving for the new velocity potential at the wall. Thus, using Eq. (25) to eliminate $\phi_{i,j,k+1}$ from Eq. (27), and using Eq. (28) to eliminate $\phi_{i,j-1,k}$ (when j=1) from the result gives

$$p_{i,k} = - \frac{\rho}{2\Delta t} \Big\{ -\phi_{i,j,k-1} - \phi_{i,j,k-1} + 2\phi_{i,j,k} +$$

$$(\frac{c\Delta t}{\Delta x})^2 [\phi_{i-1,j,k} + \phi_{i+1,j,k} - 4\phi_{i,j,k} + \phi_{i,j+1,k}$$

$$+ \phi_{i,j+1,k} + \frac{\Delta x}{\Delta t} (w_{i,k-1})] \Big\}$$

A similar equation can be obtained for the wall where $j = j_{max}$. Substituting p given by Eq. (29) into Eq. (26), combining terms, and solving for $w_{i,k+1}$ gives

$$w_{i,k+1} = \frac{\rho}{2m} \Delta t \Big\{ 2\phi_{i,j,k} - 2\phi_{i,j,k-1} +$$

$$(\frac{c\Delta t}{\Delta x})^2 [\phi_{i-1,j,k} + \phi_{i+1,1,k} - 4\phi_{i,1,k} + 2\phi_{i,2,k}$$

$$+ \frac{\Delta x}{\Delta t} w_{i,k}] \Big\}$$

$$+ 2w_{i,k} - w_{i,k-1}$$

$$- \frac{EI}{m} (\frac{\Delta t}{(\Delta x)^2})^2 \Big\{ w_{i-2,k} - 4w_{i-1,k} + 6w_{i,k} - w_{i+1,k}$$

$$+ w_{i+2,k} \Big\} \Big/ (1 + \frac{1}{m} \frac{\Delta t^2}{\Delta x} c^2) \qquad (30)$$

Equation (30) is written for the wall where $j = 1$, $0 \leq x < \alpha H$. For $x > \alpha H$, $p = 0$. Once the wall deflection is known at $k+1$, the interface conditions on the fluid are known. The calculation of the fluid velocity potential is now quite straightforward using Eq. (25) and Eq. (28) at an interface.

On the free surface of the fluid the velocity potential may be excited by an applied pressure. In this case, applying central finite differences to Eq. (24) gives

$$\phi_{k+1} = \phi_{k-1} - 2 \frac{\Delta t}{\rho} (p_b)_k \qquad (31)$$

where $(p_b)_k$ is the pressure exerted on the boundary at time step k.

B. INITIAL CONDITIONS

1. The Fluid

Since the governing partial differential equation (Eq. (19)) is second order with respect to time, two initial conditions are required; 1) the velocity potential at time zero, and 2) the first time derivative of the velocity potential, which is proportional to any initial pressure in the fluid.

2. The Wall

The beam equation (Eq. (20)) is also second order with respect to time, and two initial conditions are required; 1) the initial displacement of the wall, and 2) the initial velocity of the wall.

C. THE TIME STEP FOR NUMERICAL STABILITY

1. The Fluid

For hyperbolic systems solved using Eq. (25), the non-dimensional time step ratio

$$r = \frac{\Delta t'}{\Delta x'} \tag{32}$$

where primes denote non-dimensional quantities, must be less than 1 for numerical stability for a one-dimensional system [Ref. 3] and must be less than $\sqrt{2}/2$ for numerical stability for a two-dimensional system [Ref. 4].

Non-dimensional time t' is given by

$$t' = \frac{tc}{L} \tag{33}$$

where L is a reference length. The non-dimensional length x' is given by

$$x' = \frac{x}{L} \tag{34}$$

Substituting Eqs. (33) and (34) into Eq. (32) gives

$$t \leq \frac{\Delta x}{c\sqrt{2}} \tag{35}$$

for numerical stability of the two-dimensional system.

2. The Wall

Non-dimensionalizing the time and length coordinates in the beam equation gives [Ref. 3]

$$t' = \frac{t}{L^2} \sqrt{\frac{EI}{m}} \tag{36}$$

$$x' = \frac{x}{L} \tag{37}$$

The beam equation is parabolic in nature [Ref. 3] and requires a non-dimensional time step ratio

$$r = \frac{\Delta t'}{(\Delta x')^2} \leq \frac{1}{2} \tag{38}$$

for numerical stability. Substituting Eqs. (36) and (37) into Eq. (38) gives

$$\frac{1}{2} \geq \frac{\Delta t}{(\Delta x)^2} \sqrt{\frac{EI}{m}} \tag{39}$$

Therefore

$$\Delta t \leq \frac{(\Delta x)^2}{2} \sqrt{\frac{m}{EI}} \tag{40}$$

3. <u>The Combined System</u>

The assumption has been made that the fluid will require the smallest time step for stability. Thus, there is a restriction on the value of the parameters used in the combined analysis, in particular the physical parameters of the beam. This restriction is given by

$$\frac{1}{c\sqrt{2}} \leq \frac{\Delta x}{2} \sqrt{\frac{m}{EI}} \tag{41}$$

according to Eqs. (35) and (40).

IV. NUMERICAL METHOD OF ANALYSIS – VELOCITY FORMULATION

A. VELOCITY FORMULATION APPROACH

Kreig and Monteith [Ref. 5] have suggested a modification to the timewise finite differencing scheme outlined in Section III. This modification is described as a "velocity" formulation.

Consider their example of the one-dimensional spring mass system.

$$M\ddot{w} + Kw = 0 \tag{42}$$

Using central finite differences, Eq. (42) becomes

$$\frac{M}{(\Delta t)^2}\left\{w_{k-1} - 2w_k + w_{k+1}\right\} + Kw_k = 0 \tag{43}$$

or

$$w_{k+1} = [2 - \frac{K}{M}(\Delta t)^2]w_k - w_{k-1} \tag{44}$$

Now consider the velocity formulation

$$\begin{aligned} M\dot{v} + Kw &= 0 \\ \dot{w} &= v \end{aligned} \tag{45}$$

Defining the velocity at half time steps and the displacement at whole time steps leads to

$$\frac{M}{\Delta t} \left\{- v_{k-\frac{1}{2}} + v_{k+\frac{1}{2}}\right\} + KW_k = 0 \tag{46}$$

or

$$v_{k+\frac{1}{2}} = v_{k-\frac{1}{2}} - \frac{K}{M} \Delta t \; w_k \tag{47}$$

and

$$w_{k+1} = w_k + \Delta t \; v_{k+\frac{1}{2}} \tag{48}$$

With either formulation the time step is the same, and in the real number system without round-off error the results would be identical. However, computers do not compute in the real number system. Round-off errors are always present. Consider the term in brackets in Eq. (44)

$$[2 - \frac{K}{M} (\Delta t)^2]$$

Depending on the number of digits retained, the significance of $\frac{K}{M} (\Delta t)^2$ could be completely lost in round off.

B. FINITE DIFFERENCING SCHEME

1. The Fluid

The fluid equation derived in Section III can be written as

$$c^2\nabla^2\phi = \dot{\psi} \tag{49}$$

$$\dot{\phi} = \psi \tag{50}$$

A central finite difference scheme is proposed centering the "velocity" at whole time steps and the "displacement" at half time steps. Note, the pressure is directly computed at each whole time step since

$$p_k = - \rho\ \psi_k \tag{51}$$

The finite difference equation for the fluid is given by

$$\frac{c^2}{(\Delta x)^2} \left\{\phi_{i,j-1} + \phi_{i,j+1} - 4\phi_{i,j} + \phi_{i+1,j} + \phi_{i-1,j}\right\}_{k-\frac{1}{2}} =$$

$$\frac{1}{\Delta t} \left\{-\psi_{k-1} + \psi_k\right\}_{i,j} \tag{52}$$

where $\psi_{i,j,k}$ is the only unknown. The potential is obtained from the finite difference expression for Eq. (50)

$$\phi_{i,j,k+\frac{1}{2}} = \phi_{i,j,k-\frac{1}{2}} + \Delta t\psi_{i,j,k} \tag{53}$$

2. The Wall

Similarly, the beam equation can be written

$$EI \frac{\partial^4}{\partial x^4} w(x,t) + m\dot{v} = -p \tag{54}$$

and

$$\dot{w}(x,t) = v \tag{55}$$

For these equations a central finite difference scheme is proposed centering the velocity v at half time steps and the displacement w at whole time steps. Note that the velocity needed for the interface condition is directly computed,

$$v_{i,k-\frac{1}{2}} = \frac{1}{\Delta x} \left\{ -\phi_{i,j-1} + \phi_{i,j+1} \right\}_{k-\frac{1}{2}} \tag{56}$$

The finite difference equation for the beam is

$$\frac{EI}{(\Delta x)^4} \left\{ w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2} \right\}_k +$$

$$\frac{m}{\Delta t} \left\{ -v_{i,k-\frac{1}{2}} + v_{i,k+\frac{1}{2}} \right\} = \rho \, \psi_{i,j,k} \tag{57}$$

The unknown is $v_{i,k+\frac{1}{2}}$. From Eq. (55)

$$w_{i,k+1} = w_{i,k} + \Delta t \; v_{i,k+\frac{1}{2}} \tag{58}$$

3. The Combined System

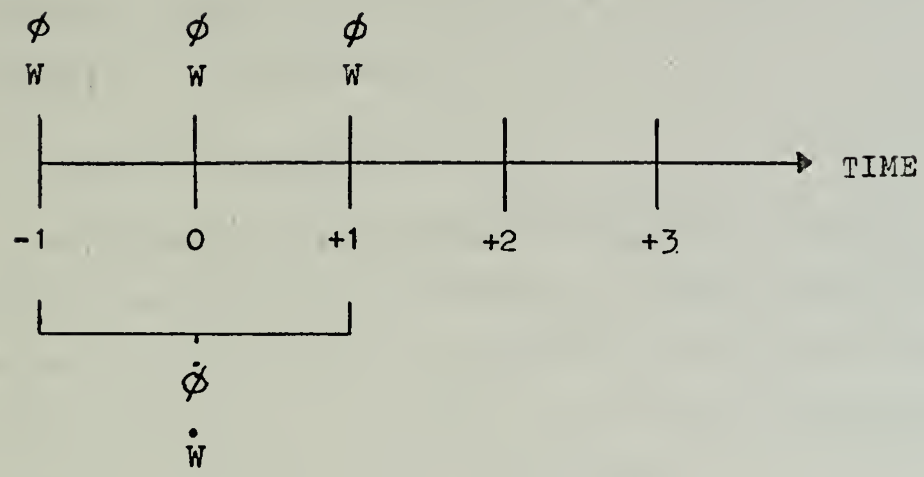
With the velocity function there is no need to solve for the beam displacement in terms of an implicit solution of the potential at the wall. In the velocity formulation the pressure is computed directly; no differencing is required. Figure 2 graphically illustrates the time step centering of both the acceleration formulation and the velocity formulation.

The solution scheme for the velocity formulation is:

1. Solve for the wall velocity, Eq. (57), based on the known pressure at the wall and the previous displacement and velocity.
2. Calculate the new wall displacement, Eq. (58), based on the new wall velocity and the previous displacement.
3. Solve for ψ, Eq. (52), based on the previous ψ, the new wall velocity (for the interface conditions) and the previous ϕ.
4. Calculate the new ϕ, Eq. (53), based on the previous ϕ and the newly calculated ψ.
5. Return to step 1 and repeat.

As in the example given by Kreig and Monteith this formulation is exactly equal to the acceleration formulation if the beam and fluid equations are considered separately. However, in the combined system the fluid equation is displaced exactly one-half time step from the fluid equation used in the acceleration formulation. In the acceleration formulation

ACCELERATION FORMULATION



VELOCITY FORMULATION

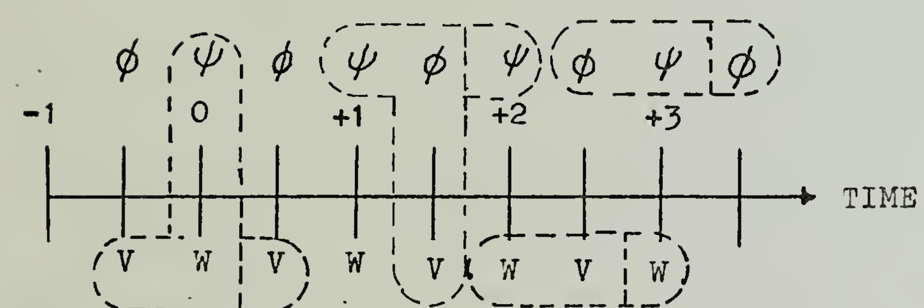


FIG. 2

the pressure depends on ϕ_{k+1} and ϕ_{k-1} whereas in the velocity formulation, it depends on $\phi_{k-\frac{1}{2}}$ and $\phi_{k+\frac{1}{2}}$. Thus, even with no round-off error there will be a difference between the two solutions. Refer to Part D for a discussion of the significance of this difference.

C. INITIAL CONDITIONS

As with the acceleration formulation a total of four initial conditions are required. However, recall that two of the four variables are defined one-half time step on either side of t = 0. They are the wall velocity and the velocity potential of ϕ in the fluid.

The actual initial conditions can be denoted as $\dot{w}_o$ and ϕ_o . The computer code assumes that

$$v_{-\frac{1}{2}} = \dot{w}_o \qquad (59)$$

and

$$\phi_{-\frac{1}{2}} = \phi_o \qquad (60)$$

D. THE TIME STEP FOR NUMERICAL STABILITY

Theoretically, since the same parameters are used in both formulations, the time step for numerical stability should be exactly the same for both formulations. In fact, this was found to be true for the beam equation and for the fluid equation when each was tested separately. However, when the systems were coupled, and the maximum theoretical

time step was used, substantial instability exhibited itself immediately in the computer results. The instability occurred at the interface and it occurred within two time steps of a pressure pulse striking a wall. A subsequent velocity was imparted to the wall which was always large enough to cause a reversal in the fluid pressure at the wall at the next time step.

A primitive but effective solution to the instability problem was obtained by reducing the time step arbitrarily to

$$\Delta t = \frac{\Delta x}{c\sqrt{3}} \qquad (61)$$

V. DESCRIPTION OF THE COMPUTER PROGRAMS

Two Fortran IV programs were written and tested. The theory behind each is outlined in Sections III and IV above. The two programs are quite similar in construction and require similar amounts of core storage. C.P.U. time required is also similar. Only one program will be described in detail, but differences between the two will be pointed out where they occur.

A. GENERAL FEATURES

The programs allow for complete flexibility in any of the possible initial conditions, both on the walls and on the fluid. At present the program allows for three walls and a variable fluid depth. The ends of each wall are considered to be clamped. The wall boundary conditions can be changed by any future user by changing six equations in the velocity formulation and twelve in the acceleration formulation. The location of these equations will be pointed out below. The program provides for a non-dimensional thirty by thirty mesh point domain, with the reference length being the length of the horizontal wall (See Fig. 1). All variables, vectors and arrays are placed in blank Common, to allow for complete flexibility in communication between subroutines. Further all vectors, and arrays are initialized to zero value before analysis begins.

Subroutines are provided to:

1. Compute constants derived from input parameters, normalize the dimensions of the problem, and compute or set initial conditions.
2. Output data obtained for each mesh point in the domain. The format is tabular.
3. Compute the beam equations.
4. Compute the fluid equations.
5. Provide for a special graphical output.

Each subroutine will be described in detail.

B. MAIN PROGRAM

1. <u>Variable Description – Velocity Formulation</u>

WA(I)	- left vertical wall displacement at k
WB(I)	- horizontal wall displacement at k
WC(I)	- right vertical wall displacement at k
VA(I)	- left vertical wall displacement at k+1
VB(I)	- horizontal wall displacement at k+1
VC(I)	- right vertical wall displacement at k+1
UA(I)	- left vertical wall velocity at k-½
UB(I)	- horizontal wall velocity at k-½
UC(I)	- right vertical wall velocity at k-½
XA(I)	- left vertical wall velocity at k+½
XB(I)	- horizontal wall velocity at k+½
XC(I)	- right vertical wall velocity at k+½
WADOT(I)	- initial left vertical wall velocity
WBDOT(I)	- initial horizontal wall velocity

WCDOT(I)	-	initial right vertical wall velocity
PRA(I)	-	initial left vertical wall pressure
PRB(I)	-	initial horizontal wall pressure
PRC(I)	-	initial right vertical wall pressure
PK(I)	-	fluid free surface applied pressure
IWA(I)	-	integer value of left vertical wall displacement
IWB(I)	-	integer value of horizontal wall displacement
IWC(I)	-	integer value of right vertical wall displacement
P(I,J)	-	value of ϕ in the fluid at $k-\frac{1}{2}$
PN(I,J)	-	value of ϕ in the fluid at $k+\frac{1}{2}$
PL(I,J)	-	value of ψ in the fluid at k-1
PRESS(I,J)	-	value of pressure in fluid at k
PHIDOT(I,J)	-	initial value of ψ in the fluid
IPRESS(I,J)	-	integer value of pressure in the fluid.

In the displacement formulation the following variables differ.

VA,VB,VC	are not used
WA,WB,WC	are values of wall displacement at k
XA,XB,XC	are values of wall displacement at k+1
UA,UB,UC	are values of wall displacement at k-1
P(I,J)	are values of ϕ at k
PN(I,J)	are values of ϕ at k+1
PL(I,J)	are values of ϕ at k-1

2. Main Program Functions

In the velocity formulation the main program accomplishes the following functions:

1. Controls calls to other subroutines.
2. Computes first value of PN(I,J) from initial conditions.
3. Saves the value of the displacement at a selected wall mesh point at selected time steps for later plotting as a time vs. displacement graph.
4. Turns off a free surface pressure disturbance at any preselected time step.

In addition the "acceleration" formulation program

5. Computes the k = 1 value of wall displacement from initial conditions.

C. SUBROUTINE INP(IER)

This subroutine

1. Reads all input data.
2. Computes constants.
3. Assigns values to initial conditions other than zero value.

1. Variable Description

RHOF	- Density of the fluid
E	- Young's Modulus
RL	- Horizontal length
RH	- Vertical height where RH $\leq$ RL
AL	- Fractional depth of fluid $0 < AL \leq 1.0$.
C	- Speed of sound in the fluid.

TH	- Thickness of beam
RHO	- Density of beam material in lbs per in^3
NH1	- Number of vertical mesh points
N1	- Number of horizontal mesh points where NH1 $\leq$ N1 = 30
DELX	- Width of mesh
DELT	- Time step
RIO	- Computed value of I, moment of inertia
NA	- Number of vertical mesh points in the fluid.
C1,C2,C3	- Constants common to both programs
C4,C5,C6,C7,C8,C9,D	- Constants used only in acceleration formulation
D1,D2,D3	- Constants used in velocity formulation
NH	- Equals NH1 - 1
NHI	- Equals NH1 - 2
N	- Equals N1 - 1
NAI	- Equals NA - 1

2. Input Cards

Only two input cards are used to input RL, RH, AL, C, TH, RHO, RHOF, E. The format will be described in Appendix A.

3. Non-Zero Initial Conditions

Any non-zero initial conditions are inserted by the user at the end of Subroutine INP.

D. SUBROUTINE OUTP(IER)

Subroutine OUTP prints out in tabular format the values of each wall mesh point displacement and the pressure at each mesh point throughout the domain.

E. SUBROUTINE WALL(IER)

Subroutine WALL computes the values of the wall velocity and displacement in the velocity formulation and the wall displacement in the acceleration formulation. If a future user desired to change the boundary conditions of the beam equation, then any equation with a computed answer stored in a variable indexed with a 2, N, or NH would have to be changed. In the velocity formulation only the equations for VA(2), VA(NH), VB(2), VB(N), VC(2), VC(NH) need to be changed. In the acceleration formulation these equations are for XA(2), XA(NH), XB(2), XB(N), XC(2), XC(NH) and recall that these equations must also be changed in the main program.

F. SUBROUTINE FLUID(IER)

Subroutine FLUID computes the value of ψ and ϕ at k and k+½ respectively in the velocity formulation and computes the values of ϕ at k+1 in the acceleration formulation. Pressure at each mesh point is also computed in FLUID.

G. SUBROUTINE GRAPHO(IER)

Subroutine GRAPHO is an attempt to make some sense out of the possible 1900 numerical values calculated at each time step. The subroutine locates an absolute maximum value

for wall displacement and fluid pressure and from these values normalizes the entire domain to integer values from -99 to +100. This information is then presented in a format quite similar to Fig. (1). An integer value is placed at each mesh point in the domain.

VI. SOLUTIONS

A. CHECK CASES

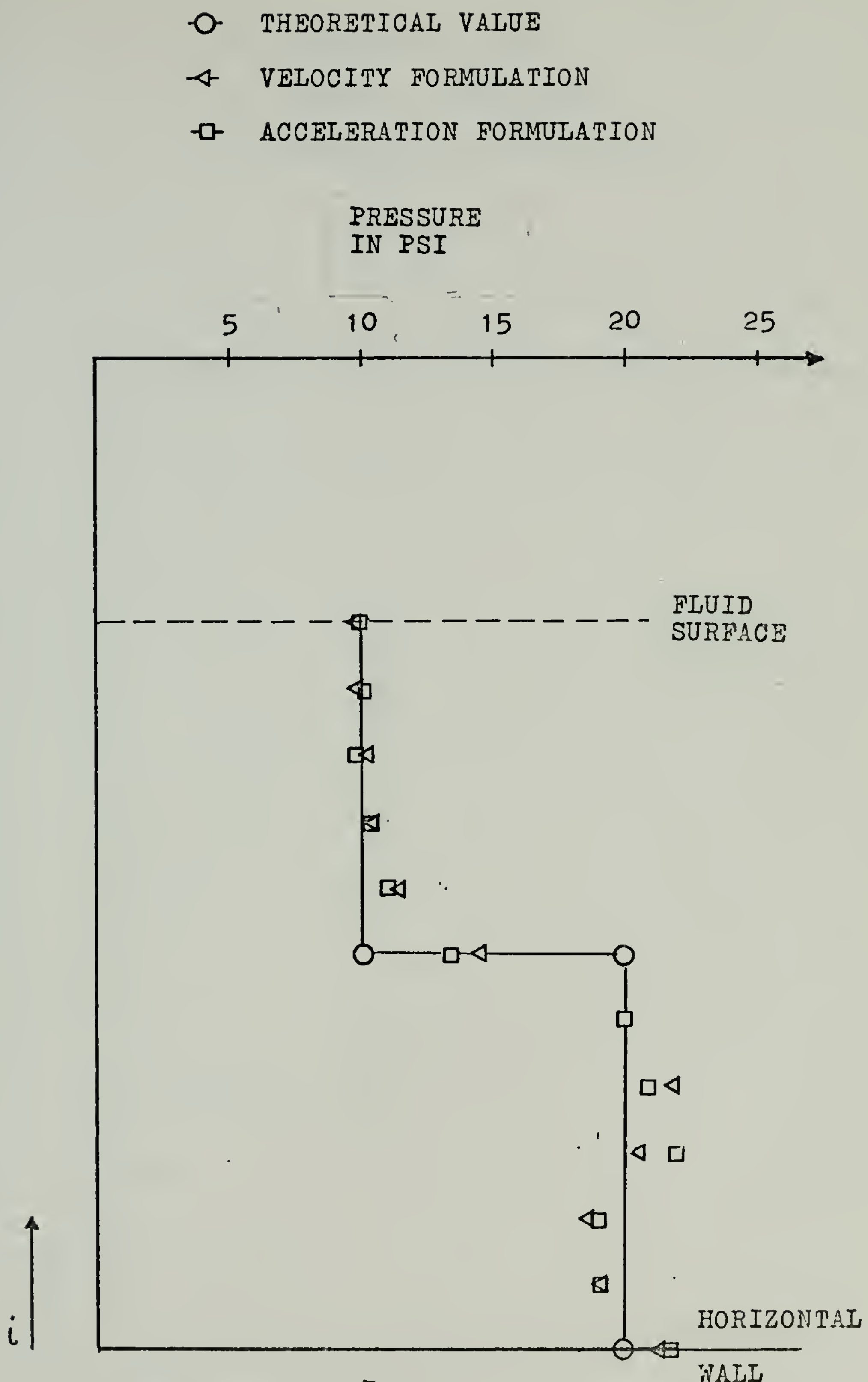
In order to determine if the programs correctly modeled the system described above, several test cases were run on each formulation.

1. The Fluid

By controlling the subroutines called by the main program, the user is able selectively to compute results for the fluid alone or the wall alone. One test case for the fluid alone was a step pressure disturbance applied uniformly over the entire upper surface for a finite time interval. The walls were treated as rigid. This test showed that the pulse propagation and reflection behaved as expected in both formulations. The uniform pulse traveled uniformly through the fluid. The value of pressure doubled as expected at the horizontal rigid wall and returned uniformly to the free surface. See Figs. 3 and 4.

Another test case consisting of a point pressure disturbance at the middle of the free surface showed that two-dimensional pulse propagation in the fluid was as expected. The amplitude of the pressure decreased radially from the point source.

In both test cases the theoretical time step for stability proved to be valid in both formulations.



TIME= $.2167*10^{-3}$ sec

FIG. 3

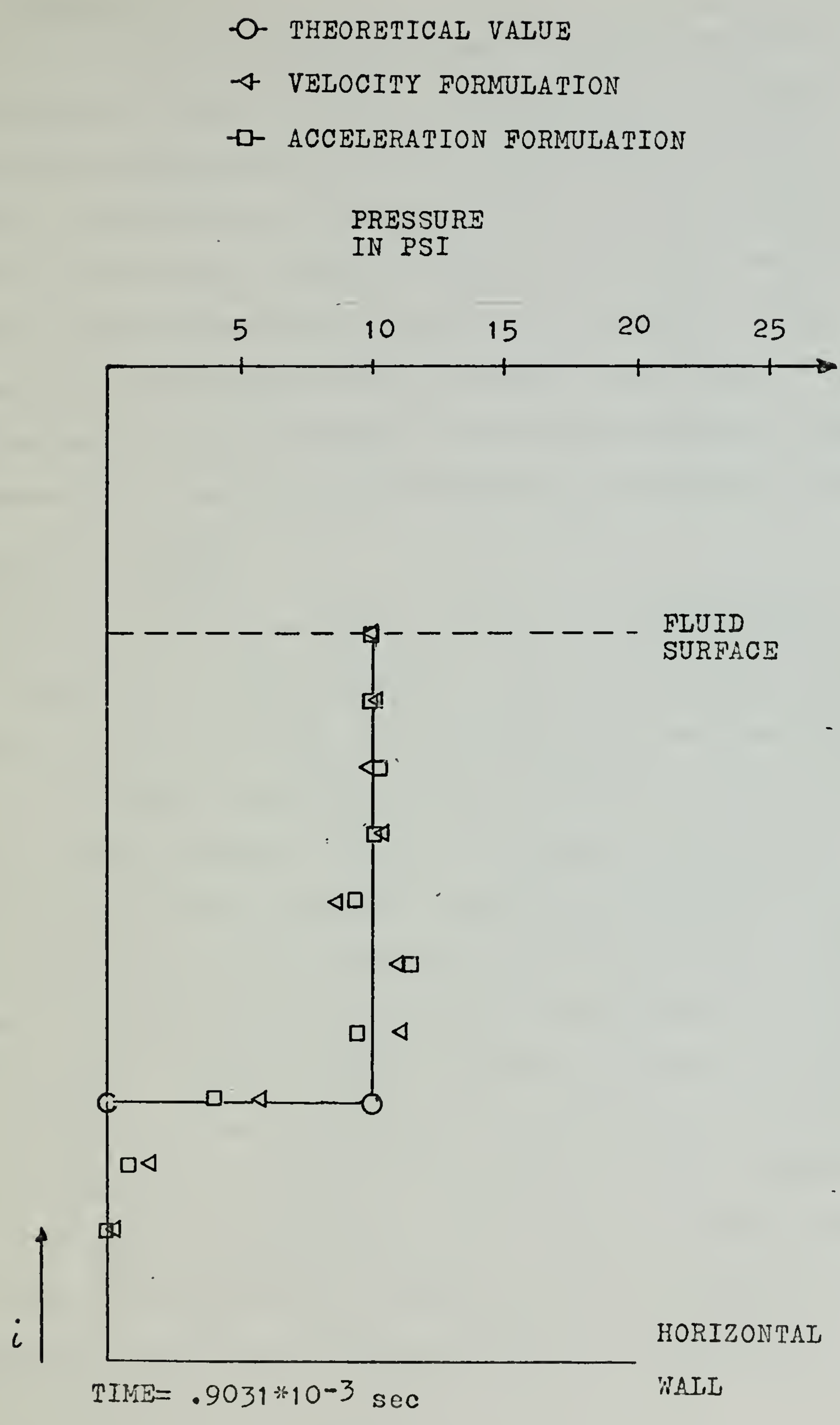

THEORETICAL VALUE
VELOCITY FORMULATION
ACCELERATION FORMULATION
PRESSURE IN PSI
5
10
15
20
25
FLUID SURFACE
i
HORIZONTAL WALL
TIME= .9031*10-3 sec

FIG. 4

2. The Wall

A free vibration analysis was conducted using the beam equation to test the wall response alone. The wall was initially displaced to conform to the first mode of vibration. Rogers [Ref. 1] gives a value of 51.3 cycles per second for the value of the frequency of the first mode of vibration for the beam parameters used in the test. The results from both formulations returned a first mode of vibration frequency of approximately 50 cycles per second. Again the theoretical time step for stability proved to be valid.

B. EXAMPLE PROBLEMS

The types of problems these two codes were developed to solve involve the combined system. The first example problem consisted of a uniform pressure pulse at the free surface of a flexible tank for 2.26 milleseconds. When a solution was first attempted using the velocity formulation, the previously mentioned instability was discovered when the theoretical time step for numerical stability of the fluid was used. Since no other solutions to this type of problem were available, first priority centered on determining the cause and cure of the instability. A reduction in the time step as discussed in Section IV solved the instability problem. The velocity formulation was run with a reduced time step, and the acceleration formulation was run using the theoretical time step. The results for the displacement

of the center of the horizontal wall are shown in Fig. (5). The dominant frequency of vibration is approximately 12 cycles per second for both formulations. There is a small amplitude discrepency of about 5% between the two formulations. The resulting vibration appeared to involve all modes in the beam.

The acceleration formulation was also used to solve the same problem with a uniform pulse held for .226 milleseconds or $1/10^{th}$ the previous example. Figure (5) again denotes a frequency of approximately 12 cycles per second, and the amplitude is proportional to the previous case. Maximum amplitude is about one-tenth that of the previous case.

It is interesting to note that if the fluid is considered as incompressible, and its mass is added to the mass of the beam, the frequency of free vibration as given by Rogers [Ref. 2] is 11.1 cycles per second.

Another example problem was run with a pressure exerted at the two center mesh points of the free surface for approximately 450 microseconds. In this case, the first mode of vibration predominated in the horizontal wall at a frequency of 13 cycles per second. Again only minimal differences in amplitude were observed between the two formulations.

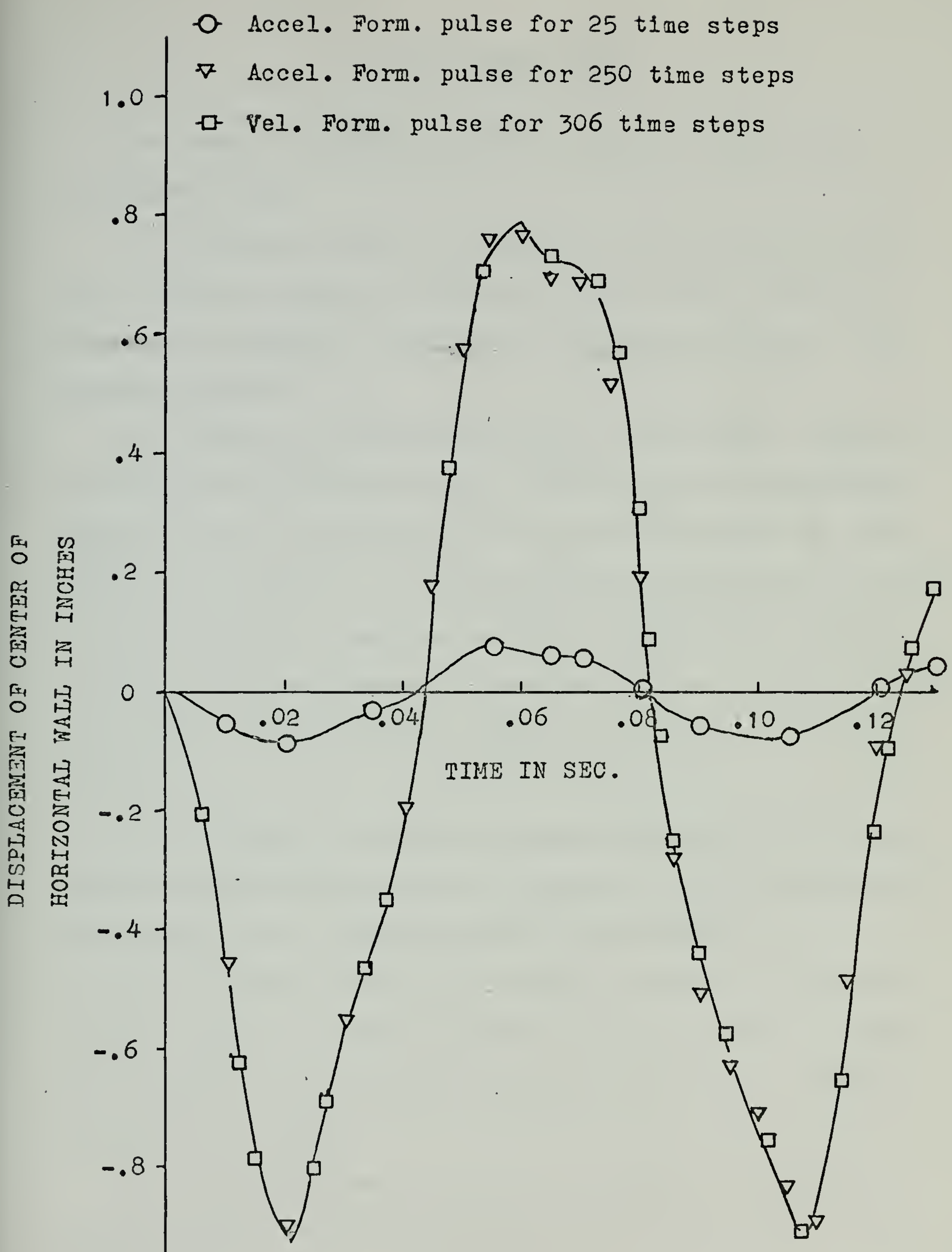

Accel. Form. pulse for 25 time steps
Accel. Form. pulse for 250 time steps
Vel. Form. pulse for 306 time steps
1.0
.8
.6
.4
.2
0
-.2
-.4
-.6
-.8
-1.0
.02
.04
.06
.08
.10
.12
TIME IN SEC.
DISPLACEMENT OF CENTER OF
HORIZONTAL WALL IN INCHES

FIG. 5

VII. CONCLUSIONS

The results obtained to date appear to support the following views:

1. The amplitude of vibration in the walls are dependent upon the length of time the free surface excitation pressure is applied. This would be expected purely from energy concepts.

2. Choice of formulation is up to the user. Each has its own virtues and its vices. The velocity formulation requires less core storage and less CPU time for the same number of time steps. (The acceleration formulation requires 15% more CPU time than the velocity formulation; however, the time step required is 19 percent smaller in the velocity formuation.) The velocity formulation would be easier to modify.

3. No claims are made as to the efficiency of either program as presently written. In fact it is estimated that the program size could be reduced considerably.

4. More work needs to be done in checking out other simple cases to prove the validity of the codes, for example, pressure pulses passing through the fluid at angles from the horizontal.

5. Either program will give a future user valuable insight about structural behavior in contact with a fluid.

APPENDIX A

USER'S INSTRUCTIONS

Input of initial conditions has already been discussed in Section V above. Actual input data is contained in two cards with the following format:

Card 1

Column	Variable	Definition	Format
Card 1:			
1-10	RL	Reference length	F10.4
11-20	RH	Vertical height	F10.4
21-30	AL	Fractional depth of fluid	F10.4
31-40	C	Velocity of sound in the fluid	F10.4
41-50	TH	Thickness of wall	F10.4
51-60	RHO	Density of wall material in $\#/in^3$	F10.4
Card 2:			
1-12	RHOF	Density of fluid in $\#\text{-}sec^2/in^4$	E12.4
13-24	E	Young's Modulus of wall material	E12.4

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     THIS IS A VELOCITY FORMULATION.
C     THIS PROGRAM SOLVES THE TWO DIMENSIONAL WALL-FLUID INTERACTION
C     PROBLEM. THE FINITE DIFFERENCE METHOD IS USED. VARIOUS INITIAL
C     CONDITIONS CAN BE EITHER READ IN AS DATA OR COMPUTED IN SUBROUTINE
C     INP.  THIS VERSION OF THE PROGRAM ALLOWS UP TO 30 BY 30 MESH
C     POINTS FOR A SQUARE DOMAIN.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DIMENSION W(401),TI(401),R(4)
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHOF,C1,C2,C3,N,NA,NH,
     5VC(30),VA(30),VB(30),D1,D2,D3,
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
CC    SET ALL VECTORS AND ARRAYS TO ZERO
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 10 I=1,30
      WA(I)=0.
      WB(I)=0.
      WC(I)=0.
      XA(I)=0.
      XB(I)=0.
      XC(I)=0.
      UA(I)=0.
      UB(I)=0.
      UC(I)=0.
      WADOT(I)=0.
      WBDOT(I)=0.
      WCDOT(I)=0.
      PRA(I)=0.
      PRB(I)=0.
      PRC(I)=0.
      PK(I)=0.
      VA(I)=0.
      VB(I)=0.
      VC(I)=0.
      IWA(I)=0.
      IWB(I)=0.
      IWC(I)=0.
      DO 10 J=1,30
      P(I,J)=0.
      PN(I,J)=0.
```

```
      PL(I,J)=0.
      PHIDOT(I,J)=0.
      PRESS(I,J)=0.
   10 IPRESS(I,J)=0.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     GO GET THE INITIAL CONDITIONS AND SIZE OF THE PROBLEM.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CALL INP(IER)
      NH1=NH+1
      N1=N+1
      NAI=NA+1
      NHI=NH-1
      NI=N-1
      NK=1
      W(1)=WB(15)
      TI(1)=0.
      DO 13 I=1,NH1
      UA(I)=WADOT(I)
      UC(I)=WCDOT(I)
      DO 13 J=1,N1
      UB(J)=WBDOT(J)
      PL(I,J)=PHIDOT(I,J)
      PN(I,J)=P(I,J)+DELT*PHIDOT(I,J)
   13 PRESS(I,J)=-RHOF*PHIDOT(I,J)
      TIME=0.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     PRINT OUT THE INITIAL CONDITIONS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CALL OUTP(IER)
      CALL GRAPHO(IER)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      START TIME STEP LOOP.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 100 K=1,15000
      RK=K
      TIME=DELT*RK
   15 CALL WALL(IER)
   19 DO 20 I=2,N
      WB(I)=XB(I)
   20 UB(I)=VB(I)
      DO 21 I=2,NH
      WA(I)=XA(I)
```

```
      WC(I)=XC(I)
      UA(I)=VA(I)
   21 UC(I)=VC(I)
      DO 22 I=1,NAI
      DO 22 J=1,N1
   22 P(I,J)=PN(I,J)
   16 CALL FLUID(IER)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     DC WE WANT TO TURN OFF PERTURBATION YET
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF(K.NE.250) GO TO 30
      I=NAI
      DO 40 J=1,N1
   40 PK(J)=0.
   30 CONTINUE
   98 CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     SELECT WHICH TIME STEP OR STEPS YOU WANT A CUTPUT FOR.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      KJ=K-(K/5000)*5000
      IF (KJ.NE.0) GO TO 99
   18 CALL GRAPHO(IER)
   99 CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     PICK OUT PERIODIC WALL INFO FOR LATER GRAPH AND STORE IN W(I).
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      KJ=K-(K/50)*50
      IF (KJ.NE.0) GO TO 100
      NK=NK+1
      W(NK)=WB(15)
      TI(NK)=TIME
  100 CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     PARAMETERS FOR GRAPH
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      R(3)= .9
      R(4)=-.9
      R(2)=0.
      R(1)=TIME
      WRITE (6,4001)
```

```
4001 FORMAT ('1',/)                                                        CCC01450
     CALL UTPLOT(TI,W,NK,R,1,0)                                            CCC01460
     WRITE (6,4001)                                                        CCC01470
     STOP                                                                  CCC01480
     END                                                                   CCC01490
```

```
      SUBROUTINE INP(IER)                                               CCC01520
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC01530
C                                                                       CCC01540
C     ANY UPPER SURFACE PERTURBATION IS STARTED IN THIS                 CCC01550
C     SUBROUTINE.                                                       CCC01560
C                                                                       CCC01570
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC01580
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),    CCC01590
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),      CCC01600
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,   CCC01610
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHOF,C1,C2,C3,N,NA,NH,   CCC01620
     5VC(30),VA(30),VB(30),D1,D2,D3,                                    CCC01630
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9   CCC01640
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC01650
C                                                                       CCC01660
C     READ  PARAMETERS OF THE PROBLEM                                   CCC01670
C                                                                       CCC01680
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC01690
      READ (5,1001) RL,RH,AL,C ,TH,RHO                                  CCC01700
      READ (5,1002) RHOF,E                                              CCC01710
 1001 FORMAT (6F10.4)                                                   CCC01720
 1002 FORMAT(2E12.4)                                                    CCC01730
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC01740
C                                                                       CCC01750
C     COMPUTE CONSTANTS AND TIME STEP. ALSO NORMALIZE                   CCC01760
C     PARAMETERS OF THE PROBLEM                                         CCC01770
C                                                                       CCC01780
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC01790
      N=29                                                              CCC01800
      RN=N                                                              CCC01810
      DELX=RL/RN                                                        CCC01820
      DELT =DELX/(C*SQRT(3.))                                           CCC01830
      RHI=RH*RN/RL+.5                                                   CCC01840
      ALI=AL*RH*RN/RL+.5                                                CCC01850
      NH=IFIX(RHI)                                                      CCC01860
      HNEW=FLOAT(NH)*DELX                                               CCC01870
      IAL=IFIX(ALI)                                                     CCC01880
      NA=IAL                                                            CCC01890
      ALNEW=FLOAT(IAL)*DELX/HNEW                                        CCC01900
      RIO=1.*TH**3/12.                                                  CCC01910
      RHC=RHO/(32.174*12.)                                              CCC01920
      C1=E*RIO/(RHO*TH)                                                 CCC01930
      C1=C1/((DELX**2/DELT)**2)                                         CCC01940
      D1=C1/DELT                                                        CCC01950
      C2=(C*DELT/DELX)**2                                               CCC01960
      D2=RHOF*DELT/(RHO*TH)                                             CCC01970
      D3=C2/DELT                                                        CCC01980
      C3=DELX/DELT                                                      CCC01990
```

```
      C4=C1/2.
      C5=RHOF*DELT/(2.*RHO*TH)
      C6=1.+C5*C2*C3
      C7=C5/C6
       C8=C1/C6
      C9=D2/RHOF
      D=DELX**4
      D=D/(E*RIO)
      WRITE (6,2001) RL,RH,AL,HNEW,ALNEW,NH,NA,DELX,DELT,C,RHOF
      WRITE (6,2002) C1,C2,C3,C4,D
 2002 FORMAT (' ',5E12.4,/)
      WRITE (6,2003) D1,D2,D3
 2003 FORMAT(' ',3E12.4,/)
 2001 FORMAT ('1',10X,'PARAMETERS OF THE PROBLEM',///,10X,'LENGTH=', F10
     1.4,5X,'HEIGHT=',F10.4,5X,'ALPHA=',F6.4,5X,///,10X,'ACTUAL PARAMETE
     2RS',//,                      10X,'HEIGHT=',F10.4,5X,'ALPHA=',5X,F6.5,
     3//,10X,'NH=',I3,5X,'NA=',I3,///,10X,'DELX=',E12.4,10X,'DELT=',
     4E12.4,10X,'C=',F10.4,10X,'FLUID DENSITY=',E12.4,///)
      I=NA+1
      N1=N+1
      PI=3.14159
      N2=N1/2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     INPUT PERTUBATIONS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 25 J=1,N1
      PK(J)=10.
   25 PHIDOT(I,J)=-PK(J)/RHOF
      RETURN
      END
```

```
      SUBROUTINE OUTP(IER)                                               CCC02340
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),     CCC02350
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),       CCC02360
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,    CCC02370
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHOF,C1,C2,C3,N,NA,NH,    CCC02380
     5VC(30),VA(30),VB(30),D1,D2,D3,                                     CCC02390
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9    CCC02400
      WRITE (6,2002) TIME                                                CCC02410
      WRITE (6,2001) (WA(I),I=1,NH1)                                     CCC02420
      WRITE (6,2003) (WB(I),I=1,N1)                                      CCC02430
      WRITE (6,2004) (WC(I),I=1,NH1)                                     CCC02440
      WRITE (6,2005) ((PRESS(NH1+1-I,J),J=1,N1),I=1,NH1)                 CCC02450
 2001 FORMAT (5X,'LEFT VERTICAL WALL DEFLECTION',//,3(5X,10E12.4,/),/)   CCC02460
 2002 FORMAT('1','TIME =',E12.4,//)                                      CCC02470
 2003 FORMAT(5X,'HORIZONTAL WALL DEFLECTION',//,3(5X,10E12.4,/),/)       CCC02480
 2004 FORMAT(5X,'RIGHT VERTICAL WALL DEFLECTION',//,3(5X,10E12.4,/),/)   CCC02490
 2005 FORMAT(5X,'PRESSURE IN THE FLUID',//,90(5X,10E12.4,/),/)           CCC02500
      RETURN                                                             CCC02510
      END                                                                CCC02520
```

```
      SUBROUTINE WALL(IER)                                              CCC02550
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),    CCC02560
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),      CCC02570
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,   CCC02580
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHOF,C1,C2,C3,N,NA,NH,   CCC02590
     5VC(30),VA(30),VB(30),D1,D2,D3,                                    CCC02600
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9   CCC02610
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC02620
C                                                                       CCC02630
C     COMPUTE ENDS FIRST.  THESE EQUATIONS CAN BE ALTERED               CCC02640
C     FOR DIFFERING BOUNDARY CONDITIONS ON THE WALL                     CCC02650
C                                                                       CCC02660
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC02670
      VA(2)=UA(2)-D1*(7.*WA(2)-4.*WA(3)+WA(4))-PRESS(2,1)*C9            CCC02680
      XA(2)=DELT*VA(2)+WA(2)                                            CCC02690
      VA(NH)=UA(NH)-D1*(7.*WA(NH)-4.*WA(NHI)+WA(NH-2))-PRESS(NH,1)*C9   CCC02700
      XA(NH)=DELT*VA(NH)+WA(NH)                                         CCC02710
      VC(2)=UC(2)-D1*(7.*WC(2)-4.*WC(3)+WC(4))-PRESS(2,N1)*C9           CCC02720
      XC(2)=DELT*VC(2)+WC(2)                                            CCC02730
      VC(NH)=UC(NH)-D1*(7.*WC(NH)-4.*WC(NHI)+WC(NH-2))-PRESS(NH,N1)*C9  CCC02740
      XC(NH)=DELT*VC(NH)+WC(NH)                                         CCC02750
      VB(2)=UB(2)-D1*(7.*WB(2)-4.*WB(3)+WB(4))-PRESS(1,2)*C9            CCC02760
      XB(2)=DELT*VB(2)+WB(2)                                            CCC02770
      VB(N)=UB(N)-D1*(7.*WB(N)-4.*WB(NI)+WB(N-2))-PRESS(1,N)*C9         CCC02780
      XB(N)=DELT*VB(N)+WB(N)                                            CCC02790
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC02800
C                                                                       CCC02810
C     COMPUTE INTERIOR WALL POINTS FOR HORIZONTAL WALL.                  CCC02820
C                                                                       CCC02830
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC02840
      DO 10 I=3,NI                                                      CCC02850
      VB(I)=UB(I)-D1*(WB(I-2)-4.*WB(I-1)+6.*WB(I)-4.*WB(I+1)            CCC02860
     1+WB(I+2))-PRESS(1,I)*C9                                           CCC02870
   10 XB(I)=DELT*VB(I)+WB(I)                                            CCC02880
      DO 11 I=3,NHI                                                     CCC02890
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC02900
C                                                                       CCC02910
C      IS WALL POINT EQUAL TO FLUID LEVEL.                              CCC02920
C                                                                       CCC02930
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC02940
      IF (I.NE.NAI) GO TO 13                                            CCC02950
      VA(I)=UA(I)-D1*(WA(I-2)-4.*WA(I-1)+6.*WA(I)-4.*WA(I+1)+WA(I+2))   CCC02960
     1 -PK(1)*C9                                                        CCC02970
      VC(I)=UC(I)-D1*(WC(I-2)-4.*WC(I-1)+6.*WC(I)-4.*WC(I+1)+WC(I+2))   CCC02980
     1 -PK(N1)*C9                                                       CCC02990
      GO TO 12                                                          CCC03000
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC03010
C                                                                       CCC03020
```

```
C      IS WALL POINT ABOVE FLUID LEVEL
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   13 IF (I.LT.NAI) GO TO 14
      VA(I)=UA(I)-D1*(WA(I-2)-4.*WA(I-1)+6.*WA(I)-4.*WA(I+1)+WA(I+2))
      VC(I)=UC(I)-D1*(WC(I-2)-4.*WC(I-1)+6.*WC(I)-4.*WC(I+1)+WC(I+2))
      GO TO 12
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      WALL POINT IS BELOW FLUID LEVEL
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   14 VA(I)=UA(I)-D1*(WA(I-2)-4.*WA(I-1)+6.*WA(I)-4.*WA(I+1)+WA(I+2))
     1 -PRESS(I,1)*C9
      VC(I)=UC(I)-D1*(WC(I-2)-4.*WC(I-1)+6.*WC(I)-4.*WC(I+1)
     1+WC(I+2))-PRESS(I,N1)*C9
   12 XA(I)=DELT*VA(I)+WA(I)
   11 XC(I)=DELT*VC(I)+WC(I)
      RETURN
      END
```

```
      SUBROUTINE FLUID(IER)
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHOF,C1,C2,C3,N,NA,NH,
     5VC(30),VA(30),VB(30),D1,D2,D3,
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     COMPUTE CORNERS FIRST
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      TEMP=PL(1,1)+D3*(P(2,1)+P(1,2)-2.*P(1,1))*2.
      PRESS(1,1)=-RHOF*TEMP
      PN(1,1)=P(1,1)+DELT*TEMP
      PL(1,1)=TEMP
      TEMP=PL(1,N1)+D3*2.*(P(2,N1)+P(1,N)-2.*P(1,N1))
      PRESS(1,N1)=-RHOF*TEMP
      PN(1,N1)=P(1,N1)+DELT*TEMP
      PL(1,N1)=TEMP
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     COMPUTE FLUID ALONG  HORIZONTAL WALL
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 10 J=2,N
      TEMP=PL(1,J)+D3*(P(1,J-1)+P(1,J+1)-4.*P(1,J)+2.*P(2,J)-2.*
     1DELX*UB(J))
      PRESS(1,J)=-RHOF*TEMP
      PN(1,J)=P(1,J)+DELT*TEMP
   10 PL(1,J)=TEMP
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     COMPUTE ALONG VERTICLE WALL FIRST THEN COMPUTE
C     INTERIOR POINTS.  UP TO , BUT NOT INCLUDING TOP OF FLUID.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 20 I=2,NA
      TEMP=PL(I,1)+D3*(P(I-1,1)+P(I+1,1)-4.*P(I,1)+2.*P(I,2)-
     1DELX*2.*UA(I))
      PRESS(I,1)=-RHOF*TEMP
      PN(I,1)=P(I,1)+DELT*TEMP
      PL(I,1)=TEMP
      TEMP=PL(I,N1)+D3*(P(I-1,N1)+P(I+1,N1)-4.*P(I,N1)+2.*P(I,N)-
     1DELX*2.*UC(I))
      PRESS(I,N1)=-RHOF*TEMP
      PN(I,N1)=P(I,N1)+DELT*TEMP
      PL(I,N1)=TEMP
```

```
      DO 20 J=2,N
      TEMP=PL(I,J)+D3*(P(I-1,J)+P(I+1,J)+P(I,J-1)+P(I,J+1)-4.*P(I,J))
      PRESS(I,J)=-RHOF*TEMP
      PN(I,J)=P(I,J)+DELT*TEMP
   20 PL(I,J)=TEMP
      I=NA+1
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     CCMPUT FLUID FREE SURFACE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 21 J=1,N1
      PN(I,J)=P(I,J)-DELT*PK(J)/RHOF
      PL(I,J)=-PK(J)/RHOF
   21 PRESS(I,J)=PK(J)
      RETURN
      END
```

```
      SUBROUTINE GRAPHO(IER)                                            CCC03920
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),    CCC03930
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),      CCC03940
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,   CCC03950
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHCF,C1,C2,C3,N,NA,NH,   CCC03960
     5VC(30),VA(30),VB(30),D1,D2,D3,                                    CCC03970
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9   CCC03980
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC03990
C                                                                       CCC04000
C     FIND ABSOLUTE MAX VALUE OF DISP AND PRESSURE                      CCC04010
C                                                                       CCC04020
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC04030
      TEMPW=ABS(WA(1))                                                  CCC04040
      TEMPP=ABS(PRESS(1,1))                                             CCC04050
      DO 20 I=1,NH                                                      CCC04060
      IF (TEMPW.GE.ABS(WA(I))) GO TO 2                                  CCC04070
      TEMPW=ABS(WA(I))                                                  CCC04080
    2 IF (TEMPW.GE.ABS(WC(I))) GO TO 10                                 CCC04090
      TEMPW=ABS(WC(I))                                                  CCC04100
   10 CONTINUE                                                          CCC04110
      DO 20 J=1,N1                                                      CCC04120
      IF (I.NE.1) GO TO 15                                              CCC04130
      IF(TEMPW.GE.ABS(WB(J))) GO TO 15                                  CCC04140
      TEMPW=ABS(WB(J))                                                  CCC04150
   15 IF (TEMPP.GE.ABS(PRESS(I,J))) GO TO 20                            CCC04160
      TEMPP=ABS(PRESS(I,J))                                             CCC04170
   20 CONTINUE                                                          CCC04180
      IF (TEMPW.GT.0.) GO TO 26                                         CCC04190
      IF (TEMPP.GT.0.) GO TO 27                                         CCC04200
      TEMPPM=TEMPP                                                      CCC04210
      TEMPP=1.                                                          CCC04220
      TEMPWM=TEMPW                                                      CCC04230
      TEMPW=1.                                                          CCC04240
      GO TO 29                                                          CCC04250
   26 TEMPWM=TEMPW                                                      CCC04260
      IF (TEMPP.GT.0.) GO TO 28                                         CCC04270
      TEMPPM=TEMPP                                                      CCC04280
      TEMPP=1.                                                          CCC04290
      GO TO 29                                                          CCC04300
   27 TEMPWM=TEMPW                                                      CCC04310
      TEMPW=1.                                                          CCC04320
   28 TEMPPM=TEMPP                                                      CCC04330
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC04340
C                                                                       CCC04350
C     NORMALIZE THE VALUES AT EACH MESH POINT.                          CCC04360
C                                                                       CCC04370
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC04380
   29 DO 30 I=1,NH1                                                     CCC04390
```

```
      IWA(I)=(WA(I)/TEMPW)*100.+.5
      IWC(I)=(WC(I)/TEMPW)*100.+.5
      DC 30 J=1,N1
      IF (I.NE.1) GO TO 30
      IWB(J)=(WB(J)/TEMPW)*100.+.5
   30 IPRESS(I,J)=(PRESS(I,J)/TEMPP)*100.+.5
      WRITE (6,1003) TIME,TEMPWM,TEMPPM
      DO 40 I=1,NH1
      WRITE(6,1001) IWA(NH1+1-I),(IPRESS(NH1+1-I,J),J=1,N1),IWC(NH1+1-I)
      IF (I.NE.NH-NA) GO TO 40
      WRITE(6,1005)
 1005 FORMAT ('+',5X,120('-'),/)
   40 CONTINUE
      WRITE (6,1004)
      WRITE (6,1002)(IWB(J),J=1,N1)
 1001 FORMAT (' ',I4,'|',30I4,'|',I4,/)
 1002 FORMAT (' ',5X,30I4,/)
 1003 FORMAT ('1',5X,'TIME =',E12.4,10X,'MAX WALL DEFLECTION =',
     11PE12.4,10X,'MAX PRESSURE IN THE FLUID =',1PE12.4,///)
 1004 FORMAT (' ',5X,120('_'),/)
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C     THIS IS AN ACCELERATION FORMULATICN                                 C
C     THIS PROGRAM SOLVES THE TWO DIMENSIONAL WALL-FLUID INTERACTION      C
C     PROBLEM. THE FINITE DIFFERENCE METHOD IS USED. VARIOUS INITIAL      C
C     CONDITIONS CAN BE EITHER READ IN AS DATA CR COMPUTED IN SUBROUTINE  C
C     INP.  THIS VERSION OF THE PROGRAM ALLOWS UP TO 30 BY 30 MESH        C
C     PCINTS FOR A SQUARE DOMAIN.                                         C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DIMENSION W(401),TI(401),R(4)
      COMMCN WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHCF,C1,C2,C3,N,NA,NH,
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9
C
CC    SET ALL VECTORS AND ARRAYS TO ZERO                                  C
C
      DC 10 I=1,30
      WA(I)=0.
      WB(I)=0.
      WC(I)=0.
      XA(I)=0.
      XB(I)=0.
      XC(I)=0.
      UA(I)=0.
      UB(I)=0.
      UC(I)=0.
      WADOT(I)=0.
      WBDOT(I)=0.
      WCDOT(I)=0.
      PRA(I)=0.
      PRB(I)=0.
      PRC(I)=0.
      PK(I)=0.
      IWA(I)=0.
      IWB(I)=0.
      IWC(I)=0.
      DO 10 J=1,30
      P(I,J)=0.
      PN(I,J)=0.
      PL(I,J)=0.
      PHIDOT(I,J)=0.
      PRESS(I,J)=0.
   10 IPRESS(I,J)=0.
C                                                                         C
C     GO GET THE INITIAL CONDITIONS AND SIZE OF THE PROBLEM.              C
```

```
C                                                                                      C
      CALL INP(IER)
      NH1=NH+1
      N1=N+1
      NAI=NA+1
      W(1)=WB(15)
      TI(1)=0.
      DO 11 I=1,NH1
      PRA(I)= RHOF*PHIDOT(I,1)
   11 PRC(I)= RHOF*PHIDOT(I,N1)
      DO 12 I=1,N1
   12 PRB(I)= RHOF*PHIDOT(1,I)
      DO 13 I=1,NH1
      DO 13 J=1,N1
   13 PRESS(I,J)=-RHOF*PHIDOT(I,J)
      TIME=0.
      CALL OUTP(IER)
      CALL GRAPHO(IER)
      XA(2)=C4*(PRA(2)*D-7.*WA(2)+4.*WA(3)-WA(4))+WA(2)+DELT*WADOT(2)
      XA(NH)=C4*(PRA(NH)*D-7.*WA(NH)+4.*WA(NH-1)-WA(NH-2))+WA(NH)+DELT*
     1WADOT(NH)
      XC(2)=C4*(PRC(2)*D-7.*WC(2)+4.*WC(3)-WC(4))+WC(2)+DELT*WCDOT(2)
      XC(NH)=C4*(PRC(NH)*D-7.*WC(NH)+4.*WC(NH-1)-WC(NH-2))+WC(NH)+DELT*
     1WCDOT(NH)
      UA(2)=XA(2)-2.*DELT*WADOT(2)
      UC(2)=XC(2)-2.*DELT*WCDOT(2)
      UA(NH)=XA(NH)-2.*DELT*WADOT(NH)
      UC(NH)=XC(NH)-2.*DELT*WCDOT(NH)
      XB(2)=WB(2)+DELT*WBDOT(2)+C4*(PRB(2)*D-7.*WB(2)+4.*WB(3)-WB(4))
      XB(N)=WB(N)+DELT*WBDOT(N)+C4*(PRB(N)*D-7.*WB(N)+4.*WB(N-1)-WB(N-2)
     1)
      UB(2)=XB(2)-2.*DELT*WBDOT(2)
      UB(N)=XB(N)-2.*DELT*WBDOT(N)
      NHI=NH-1
      DO 15 I=3,NHI
      XA(I)=WA(I)+DELT*WADOT(I)+C4*(PRA(I)*D-WA(I-2)+4.*WA(I-1)-6.*WA(I)
     1+4.*WA(I+1)-WA(I-1))
      XC(I)=WC(I)+DELT*WCDOT(I)+C4*(PRC(I)*D-WC(I-2)+4.*WC(I-1)-6.*WC(I)
     1+4.*WC(I+1)-WC(I+2))
      UA(I)=XA(I)-2.*DELT*WADOT(I)
   15 UC(I)=XC(I)-2.*DELT*WCDOT(I)
      NI=N-1
      DO 16 I=3,NI
      XB(I)=WB(I)+DELT*WBDOT(I)+C4*(PRB(I)*D-WB(I-2)+4.*WB(I-1)-6.*WB(I)
     1     +4.*WB(I+1)-WB(I+2))
   16 UB(I)=XB(I)-2.*DELT*WBDOT(I)
      PN(1,1)=P(1,1)+DELT*PHIDOT(1,1)+C2*(P(2,1)+P(1,2)-2.*P(1,1))
      PN(1,N1)=P(1,N1)+DELT*PHIDOT(1,N1)+C2*(P(2,N1)+P(1,N)-2.*P(1,N1))
```

```
      PL(1,1)=PN(1,1)-2.*DELT*PHIDOT(1,1)
      PL(1,N1)=PN(1,N1)-2.*DELT*PHIDOT(1,N1)
      DC 17 J=2,N
      PN(1,J)=P(1,J)+DELT*PHIDOT(1,J)+C2*(-4.*P(1,J)+2.*P(2,J)-C3*(XB(J)
     1 -UB(J))+P(1,J-1)+P(1,J+1))/2.
   17 PL(1,J)=PN(1,J)-2.*DELT*PHIDOT(1,J)
      DO 18 I=1,NA
      PN(I,1)=P(I,1)+DELT*PHIDOT(I,1)+C2*(-4.*P(I,1)+P(I+1,1)+P(I-1,1)
     1+2.*P(I,2)-C3*(XA(I)-UA(I)))/2.
      PN(I,N1)=P(I,N1)+DELT*PHIDOT(I,N1)+C2*(-4.*P(I,N1)+P(I+1,N1)
     1+P(I-1,N1)+2.*P(I,N)-C3*(XC(I)-UC(I)))/2.
      PL(I,1)=PN(I,1)-2.*DELT*PHIDOT(I,1)
      PL(I,N1)=PN(I,N1)-2.*DELT*PHIDOT(I,N1)
      DC 18 J=2,N
      PN(I,J)=P(I,J)+DELT*PHIDOT(I,J)+C2*(P(I-1,J)-4.*P(I,J)+P(I+1,J)
     1+P(I,J-1)+P(I,J+1))/2.
   18 PL(I,J)=PN(I,J)-2.*DELT*PHIDOT(I,J)
      I=NA+1
      DC 19 J=1,N1
      PRESS (I,J)=PK(J)
   19 PN(I,J)=-DELT*PK(J)/RHOF
      NK=1
      DC 100 K=1,50
      RK=K
      TIME=DELT*RK
      IF(K.NE.25) GO TO 30
      DC 40 J=1,N1
   40 PK(J)=0.
   30 CONTINUE
      DO 20 I=2,N
      UB(I)=WB (I)
   20 WB(I)=XB(I)
      DC 21 I=2,NH
      UA(I)=WA(I)
      UC(I)=WC(I)
      WA(I)=XA(I)
      WC(I)=XC(I)
   21 CONTINUE
      DO 22 I=1,NAI
      DC 22 J=1,N1
      PL(I,J)=P(I,J)
   22 P(I,J)=PN(I,J)
   89 CALL GRAPHO(IER)
      GO TO 91
   90 CALL WALL(IER)
   91 CONTINUE
      CALL FLUID(IER)
   92 CONTINUE
```

```
      NK=NK+1
      W(NK)=WB(15)
      TI(NK)=TIME
  100 CONTINUE
      R(3)=.001
      R(4)=-.001
      R(2)=0.
      R(1)=TIME
      WRITE (6,4001)
 4001 FORMAT ('1',/)
      CALL UTPLOT(TI,W,NK,R,1,0)
      WRITE (6,4001)
      STOP
      END
```

```
      SUBROUTINE INP(IER)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     AN UPPER SURFACE PRESSURE IS DENOTED BY PK(J).  IT IS
C     TURNED ON IN THIS SUBROUTINE AND TURNED OFF IN MAIN
C     PROGRAM  IN SIDE THE THE K DO LOOP
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHCF,C1,C2,C3,N,NA,NH,
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9
      READ (5,1001) RL,RH,AL,C ,TH,RHO
 1001 FORMAT (6F10.4)
      READ (5,1002) RHOF,E
 1002 FORMAT(2E12.4)
      N=29
      RN=N
      DELX=RL/RN
      DELT=DELX/(C*SQRT(2.))
      RHI=RH*RN/RL+.5
      ALI=AL*RH*RN/RL+.5
      NH=IFIX(RHI)
      HNEW=FLOAT(NH)*DELX
      IAL=IFIX(ALI)
      NA=IAL
      ALNEW=FLOAT(IAL)*DELX/HNEW
      RIO=1.*TH**3/12.
      RHO=RHO/(32.174*12.)
      C1=E*RIO/(RHO*TH)
      C1=C1/((DELX**2/DELT)**2)
      C2=(C*DELT/DELX)**2
      C3=DELX/DELT
      C4=C1/2.
      C4=0.
      C5=RHOF*DELT/(2.*RHO*TH)
      C6=1.+C5*C2*C3
      C7=C5/C6
       C8=C1/C6
      C9=1./C6
      D=DELX**4
      D=D/(E*RIO)
      WRITE (6,2001) RL,RH,AL,HNEW,ALNEW,NH,NA,DELX,DELT,C,RHOF
      WRITE (6,2002) C1,C2,C3,C4,D
 2002 FORMAT (' ',5E12.4,/)
 2001 FORMAT ('1',10X,'PARAMETERS OF THE PROBLEM',///,10X,'LENGTH=', F10
     1.4,5X,'HEIGHT=',F10.4,5X,'ALPHA=',F6.4,5X,///,10X,'ACTUAL PARAMETE
```

```
2RS',//,                       10X,'HEIGHT=',F10.4,5X,'ALPHA=',5X,F6.5,
3//,10X,'NH=',I3,5X,'NA=',I3,///,10X,'DELX=',E12.4,10X,'DELT=',
4E12.4,10X,'C=',F10.4,10X,'FLUID DENSITY=',E12.4,///)
 I=NA+1
 N1=N+1
 PI=3.14159
 N2=N1/2
 DO 25 J=1,N1
 PK(J)=10.
25 PHIDOT(I,J)=-PK(J)/RHOF
 RETURN
 END
```

```
      SUBROUTINE OUTP(IER)
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHCF,C1,C2,C3,N,NA,NH,
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9
      WRITE (6,2002) TIME
      WRITE (6,2001) (WA(I),I=1,NH1)
      WRITE (6,2003) (WB(I),I=1,N1)
      WRITE (6,2004) (WC(I),I=1,NH1)
      WRITE (6,2005) ((PRESS(NH1+1-I,J),J=1,N1),I=1,NH1)
 2001 FORMAT (5X,'LEFT VERTICAL WALL DEFLECTION',//,3(5X,10E12.4,/),/)
 2002 FORMAT('1','TIME =',E12.4,//)
 2003 FORMAT(5X,'HORIZONTAL WALL DEFLECTION',//,3(5X,10E12.4,/),/)
 2004 FORMAT(5X,'RIGHT VERTICAL WALL DEFLECTION',//,3(5X,10E12.4,/),/)
 2005 FORMAT(5X,'PRESSURE IN THE FLUID',//,90(5X,10E12.4,/),/)
      RETURN
      END
```

```
      SUBROUTINE WALL(IER)
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHCF,C1,C2,C3,N,NA,NH,
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9
      XA(2)=C7*(2.*P(2,1)-2.*PL(2,1)+C2*(P(1,1)+P(3,1)-4.*P(2,1)+
     12.*P(2,2)+C3*UA(2)))+C9*(2.*WA(2)-UA(2))-C8*(7.*WA(2)-4.*WA(3)+
     1WA(4))
      XC(2)=C7*(2.*P(2,N1)-2.*PL(2,N1)+C2*(P(1,N1)+P(3,N1)-4.*P(2,N1)+
     12.*P(2,N)+C3*UC(2)))+C9*(2.*WC(2)-UC(2))-C8*(7.*WC(2)-4.*WC(3)+
     2WC(4))
      XC(NH)=C7*(2.*P(NH,N1)-2.*PL(NH,N1)+C2*(P(NH1,N1)+P(NHI,N1)-
     14.*P(NH,N1)+2.*P(NH,N)+C3*UC(NH)))+C9*(2.*WC(NH)-UC(NH))-C8*
     2(7.*WC(NH)-4.*WC(NHI)+WC(NH-2))
      XA(NH)=C7*(2.*P(NH,1)-2.*PL(NH,1)+C2*(P(NH1,1)+P(NHI,1)-4.*P(NH,1)
     2+2.*P(NH,2)+C3*UA(NH)))+C9*(2.*WA(NH)-UA(NH))-C8*(7.*WA(NH)-4.*
     2WA(NHI)+WA(NH-2))
      XB(2)=C7*(2.*P(1,2)-2.*PL(1,2)+C2*(P(1,1)+P(1,3)-4.*P(1,2)
     1+2.*P(2,2)+C3*UB(2)))+C9*(2.*WB(2)-UB(2))-C8*(7.*WB(2)-4.*WB(3)
     2+WB(4))
      XB(N)=C7*(2.*P(1,N)-2.*PL(1,N)+C2*(P(1,N1)+P(1,NI)-4.*P(1,N)
     2+2.*P(2,N)+C3*UB(N)))+C9*(2.*WB(N)-UB(N))-C8*(7.*WB(N)-4.*WB(NI)
     3+WB(N-2))
      DO 10 I=3,NI
   10 XB(I)=C7*(2.*P(1,I)-2.*PL(1,I)+C2*(P(1,I-1)+P(1,I+1)-4.*P(1,I)
     1+2.*P(2,I)+C3*UB(I)))+C9*(2.*WB(I)-UB(I))-C8*(WB(I-2)-4.*WB(I+1)
     2+6.*WB(I)-4.*WB(I-1)+WB(I+2))
      DO 11 I=3,NHI
      IF (I.NE.NAI) GO  TO 12
      XA(I)=2.*WA(I)-UA(I)-C1*(WA(I-2)-4.*WA(I-1)+6.*WA(I)-4.*WA(I+1)
     1+WA(I+2)+PK(1)*D)
      XC(I)=2.*WC(I)-UC(I)-C1*(WC(I-2)-4.*WC(I-1)+6.*WC(I)-4.*WC(I+1)
     1+WC(I+2)+PK(N1)*D)
      GO TO 11
   12 IF (I.LE.NAI) GO TO 13
      XA(I)=2.*WA(I)-UA(I)-C1*(WA(I-2)-4.*WA(I-1)+6.*WA(I)-4.*WA(I+1)
     1 +WA(I+2))
      XC(I)=2.*WC(I)-UC(I)-C1*(WC(I-2)-4.*WC(I-1)+6.*WC(I)-4.*WC(I+1)
     1 +WC(I+2))
      GO TO 11
   13 XA(I)=C7*(2.*P(I,1)-2.*PL(I,1)+C2*(P(I-1,1)+P(I+1,1)-4.*P(I,1)
     1+2.*P(I,2)+C3*UA(I)))+C9*(2.*WA(I)-UA(I))-C8*(WA(I-2)-4.*WA(I-1)
     2+6.*WA(I)-4.*WA(I+1)+WA(I+2))
      XC(I)=C7*(2.*P(I,N1)-2.*PL(I,N1)+C2*(P(I-1,N1)+P(I+1,N1)
     2-4.*P(I,N1)+2.*P(I,N)+C3*UC(I)))+C9*(2.*WC(I)-UC(I))-C8*(WC(I-2)
     2-4.*WC(I-1)+6.*WC(I)-4.*WC(I+1)+WC(I+2))
   11 CONTINUE
```

RETURN
END

```
      SUBROUTINE FLUID(IER)
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHOF,C1,C2,C3,N,NA,NH,
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9
      PN(1,1)=2.*P(1,1)-PL(1,1)+(2.*P(2,1)+2.*P(1,2)-4.*P(1,1))*C2
      PN(1,N1)=2.*P(1,N1)-PL(1,N1)+(2.*P(1,N)+2.*P(2,N1)-4.*P(1,N1))*C2
      PRESS(1,1)=-RHOF*(PN(1,1)-PL(1,1))/(2.*DELT)
      PRESS(1,N1)=-RHOF*(PN(1,N1)-PL(1,N1))/(2.*DELT)
      DO 10 J=2,N
      PN(1,J)=2.*P(1,J)-PL(1,J)+(P(1,J-1)+P(1,J+1)-4.*P(1,J)+2.*P(2,J)
     1-C3*(XB(J)-UB (J)))*C2
   10 PRESS(1,J)=-RHOF*(PN(1,J)-PL(1,J))/(2.*DELT)
      DO 20 I=2,NA
      PN(I,1)=2.*P(I,1)-PL(I,1)+(2.*P(I,2)-C3*(XA(I)-UA(I))-4.*P(I,1)+
     1P(I+1,1)+P(I-1,1))*C2
      PN(I,N1)=2.*P(I,N1)-PL(I,N1)+(2.*P(I,N)-C3*(XC(I)-UC(I))-4.*P(I,N1
     1)+P(I+1,N1)+P(I-1,N1))*C2
      PRESS(I,1)=-RHOF*(PN(I,1)-PL(I,1))/(2.*DELT)
      PRESS(I,N1)=-RHOF*(PN(I,N1)-PL(I,N1))/(2.*DELT)
      DO 20 J=2,N
      PN(I,J)=2.*P(I,J)-PL(I,J)+(P(I,J-1)+P(I,J+1)-4.*P(I,J)+P(I-1,J)+
     1P(I+1,J))*C2
   20 PRESS(I,J)=-RHOF*(PN(I,J)-PL(I,J))/(2.*DELT)
      I=NA+1
      DO 21 J=1,N1
      PN(I,J)=PL(I,J)-2.*DELT*PK(J)/RHOF
   21 PRESS(I,J)=PK(J)
      RETURN
      END
```

```
      SUBROUTINE GRAPHO(IER)
      COMMON WA(30),WB(30),WC(30),WADOT(30),WBDOT(30),XA(30),XB(30),
     1XC(30),UA(30),UB(30),UC(30),PRA(30),PRB(30),PRC(30),IWA(30),
     2IWB(30),IWC(30),P(30,30),PN(30,30),PL(30,30),PHIDOT(30,30),NAI,
     3PRESS(30,30),IPRESS(30,30),DELX,DELT,RHO,RHOF,C1,C2,C3,N,NA,NH,
     4WCDOT(30),NH1,N1,HNEW,TIME,NHI,NI,K,C4,D ,PK(30),C5,C6,C7,C8,C9
      TEMPW=ABS(WA(1))
      TEMPP=ABS(PRESS(1,1))
      DO 20 I=1,NH
      IF (TEMPW.GE.ABS(WA(I))) GO TO 2
      TEMPW=ABS(WA(I))
    2 IF (TEMPW.GE.ABS(WC(I))) GO TO 10
      TEMPW=ABS(WC(I))
   10 CONTINUE
      DO 20 J=1,N1
      IF (I.NE.1) GO TO 15
      IF(TEMPW.GE.ABS(WB(J))) GO TO 15
      TEMPW=ABS(WB(J))
   15 IF (TEMPP.GE.ABS(PRESS(I,J))) GO TO 20
      TEMPP=ABS(PRESS(I,J))
   20 CONTINUE
      IF (TEMPW.GT.0.) GO TO 26
      IF (TEMPP.GT.0.) GO TO 27
      TEMPPM=TEMPP
      TEMPP=1.
      TEMPWM=TEMPW
      TEMPW=1.
      GO TO 29
   26 TEMPWM=TEMPW
      IF (TEMPP.GT.0.) GO TO 28
      TEMPPM=TEMPP
      TEMPP=1.
      GO TO 29
   27 TEMPWM=TEMPW
      TEMPW=1.
   28 TEMPPM=TEMPP
   29 DO 30 I=1,NH1
      IWA(I)=(WA(I)/TEMPW)*100.+.5
      IWC(I)=(WC(I)/TEMPW)*100.+.5
      DO 30 J=1,N1
      IF (I.NE.1) GO TO 30
      IWB(J)=(WB(J)/TEMPW)*100.+.5
   30 IPRESS(I,J)=(PRESS(I,J)/TEMPP)*100.+.5
      WRITE (6,1003) TIME,TEMPWM,TEMPPM
      DO 40 I=1,NH1
      WRITE(6,1001) IWA(NH1+1-I),(IPRESS(NH1+1-I,J),J=1,N1),IWC(NH1+1-I)
      IF (I.NE.NH-NA) GO TO 40
      WRITE(6,1005)
```

```
1005 FORMAT ('+',5X,120('¬'),/)
  40 CONTINUE
     WRITE (6,1004)
     WRITE (6,1002)(IWB(J),J=1,N1)
1001 FORMAT (' ',I4,'|',30I4,'|',I4,/)
1002 FORMAT (' ',5X,30I4,/)
1003 FORMAT ('1',5X,'TIME =',E12.4,10X,'MAX WALL DEFLECTION =',
    11PE12.4,10X,'MAS PRESSURE IN THE FLUID =',1PE12.4,///)
1004 FORMAT (' ',5X,120('_'),/)
     RETURN
     END
```

BIBLIOGRAPHY

1. The Boeing Company, Technical Proposal D-162-10294-1, Hydraulic Ram, 18 September 1970.

2. Rogers, G.L., Dynamics of Framed Structures, Wiley, 1950.

3. Crandall, S.H., Engineering Analysis, McGraw-Hill, 1956.

4. Ames, W.F., Numerical Methods for Partial Differential Equations, Barnes and Noble, 1969.

5. Air Force Flight Dynamics Laboratory Technical Report 71-79, A Large Deflection Transient Analysis of Arbitrary Shells Using Finite Differences, by Kreig, R.D., and Monteith, H.C., One of several reports in a volume titled Computer Oriented Analysis of Shell Structures, June 1971.

INITIAL DISTRIBUTION LIST

		No. of Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey California 93940	2
3.	Chairman, Code 57 Department of Aeronautics Naval Postgraduate School Monterey California 93940	2
4.	Asst. Professor R. Ball, Code 57 Ba Department of Aeronautics Naval Postgraduate School Monterey California 93940	1
5.	LT. John Charles Bitzberger, USN 26 Catspaw Cape Coronado, California 92118	1

Thesis
B54537 Bitzberger
c.1
153256
Two dimensional analysis of fluid-structure interaction by method of finite differences - hydraulic ram, the fuel tank problem.